The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method

Article *in* European Journal of Operational Research · February 2010

DOI: 10.1016/j.ejor.2009.01.022 · Source: OAI

3 authors, including:

Cristián E. Cortés
University of Chile
62 PUBLICATIONS **877** CITATIONS
SEE PROFILE

Claudio Contardo
Group for Research in Decision Analysis
9 PUBLICATIONS **235** CITATIONS
SEE PROFILE

European Journal of Operational Research 200 (2010) 711–724

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method

Cristián E. Cortés [a], Martín Matamala [b], Claudio Contardo [b,*]

[a] Department of Civil Engineering, Universidad de Chile, Av. Blanco Encalada 2002, 5th Floor, Santiago, Chile
[b] Department of Mathematical Engineering, Universidad de Chile, Av. Blanco Encalada 2120, 5th Floor, Santiago, Chile

ARTICLE INFO

Article history:
Received 5 October 2006
Accepted 18 January 2009
Available online 25 January 2009

Keywords:
Pickup and delivery
Dial-a-ride
Branch-and-bound
Branch-and-cut
Benders decomposition
Mixed-integer programming

ABSTRACT

In this paper, a strict formulation of a generalization of the classical pickup and delivery problem is presented. Here, we add the flexibility of providing the option for passengers to transfer from one vehicle to another at specific locations. As part of the mathematical formulation, we include transfer nodes where vehicles may interact interchanging passengers. Additional variables to keep track of customers along their route are considered. The formulation has been proven to work correctly, and by means of a simple example instance, we conclude that there exist some configurations in which a scheme allowing transfers results in better quality optimal solutions. Finally, a solution method based on Benders decomposition is addressed. We compare the computational effort of this application with a straight branch and bound strategy; we also provide insights to develop more efficient set partitioning formulations and associated algorithms for solving real-size problems.

1. Introduction

The pickup and delivery problem (PDP), as defined in the specialized literature, has been one of the most studied network logistic problems in the last decade (Desrosiers et al., 1995; Savelsbergh and Sol, 1995; Desrochers et al., 1988). The PDP can be conceptually described as finding the optimal way of assigning a set of transportation requests to a fleet of vehicles (initially located at several depots), by minimizing a specific purpose objective function, subject to a variety of constraints. The objective function may include components such as operational costs, number of vehicles, customer's level of service, and so on.

A transportation request consists of picking up a certain number of passengers from a predetermined pickup location and dropping them off at a predetermined delivery location. In the version that includes time windows constraints, normally both locations must be reached by the assigned vehicle within a specified time interval. The fundamental formulations of the PDP are mostly static (in which the demand for the service is known in advance), however some authors have explored the dynamic problem to be solved in real time, without providing a strict optimization formulation for such a scheme (Roy et al., 1984; Psaraftis, 1988; Madsen et al., 1995; Gendreau et al., 1998).

In most static applications, the typical objective functions to be minimized are, in first place, the fleet size needed to satisfy the demand with a determined level of service, and next, the total distance traveled by all vehicles, provided that the fleet size is fixed (Desrosiers et al., 1995). In the case of passenger movements, instead of the distance attribute, a more reasonable objective function should include both the total waiting and travel time of all passengers, combined with some measure of the operational cost for running the system, weighted differently.

In the literature, the passenger PDP has to be solved for scheduling/routing the widely known dial-a-ride or demand responsive transit systems implemented all over the world (Black, 1995). Most traditional formulations consider fixed fleet size and homogeneous vehicles with limited passenger capacity (normally able to accommodate about 5–6 passengers).

The static PDP constraints can be classified as visiting constraints (each pickup and delivery has to be visited exactly once), vehicle capacity constraints, depot constraints, coupling constraints (stating that for a given request the same vehicle must visit the pickup and delivery stops), precedence constraints, resource constraints on the availability of drivers and vehicles, and time window constraints to be satisfied at each stop, if time windows are explicitly defined (Desrosiers et al., 1995).

Standard PDP is economically most suited to small fleets and low levels of demand, and that multi-hop may be best suited to situations where demand is higher. That explains in part why most PDP applications have been developed to solve problems oriented to the service of small communities or passengers with specific requirements (elderly, disabled).

As reviewed in detail in the next section, in the literature various authors have conceived integrated schemes involving fixed

* Corresponding author.
E-mail addresses: ccortes@ing.uchile.cl (C.E. Cortés), mmatamal@dim.uchile.cl (M. Matamala), ccontard@gmail.com (C. Contardo).

0377-2217/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2009.01.022

route systems and reroutable services to overcome such limitations of the PDP scheme, allowing the possibility for passengers to transfer from one vehicle to another before completing his(her) trip. These schemes that add flexibility to the operation could improve passenger demand, and improve the overall productivity by taking advantage of the interaction between vehicles at specific transfer points. The drawback of such systems is the extra disutility on passengers due to the transfer operation (involving extra waiting time and the discomfort just for the fact of performing the transfer).

Therefore, what is really important is to quantify the trade off between the advantage of adding flexibility (by relaxing the coupling constraints) and the extra disutility due to the addition of transfers to the operational scheme. Obviously, if the added flexibility significantly reduced the total passenger waiting and travel times, one could try to find favorable conditions for implementing a system allowing transfers instead of a rigid PDP including hard coupling constraints into the formulation.

This research was motivated by the necessity of evaluating such a compromise between these two possible PDP schemes (with and without transfer). We claim that the strictest way to find the advantages of one option over the other is to mathematically formulate the generic problem and be able to find an exact solution of it (or alternatively, a good solution close to optimality), for different demand and supply conditions. As a first stage, in this paper we present a strict arc-based formulation of the static PDP, allowing the option for passengers to transfer between vehicles, provided that the locations of the transfer points are fixed and known.

The development of a mathematically consistent arc-based formulation is the first step to start formalizing the PDP with transfers (henceforth, denoted PDPT). From this framework, more efficient set partitioning formulations (route-based) will be generated in further developments, in order to implement more efficient algorithms able to solve real-size problems. In this paper, the proposed arc-based formulation is solved with an exact solution method based upon a branch-and-cut algorithm.

The paper is structured as follows: in the next section, the typical PDP formulation and the associated solution methods are presented. In addition, we describe the most relevant approaches for PDP services including transfers. Then, in Section 3, the PDP with transfers is analytically written. In Section 4, we present a model validation, by showing an instance in which adding a transfer point results in a reduction of the ride time needed to serve the demand as compared with the classical PDP. In Section 5, an exact solution method is addressed, showing the advantages of such procedure when compared against a traditional branch-and-bound approach. In Section 6, we discuss about how the branching tree can be reduced by breaking symmetries and adding some redundant constraints. We finish in Section 7 by discussing the main conclusions of this paper and the subjects of further research.

2. The PDP: formulation, solution methods and extensions

The formulation we propose in this paper relies on the original PDP idea found in Desrochers et al. (1988), Desrosiers et al. (1995), Savelsbergh and Sol (1995). We present a mixed-integer formulation for the static multi-vehicle PDP, in which the $\{0,1\}$ variables represent the binary decision *vehicle k uses link* (i,j), and the real variables represent times and vehicle loads. We will concentrate our analysis on the static problem, leaving the dynamic case (discussed in Psaraftis, 1988; Madsen et al., 1995) as part of further research.

Since the PDP was introduced for the first time, several studies have been published, with focus on solving practical instances of the PDP, by means of several techniques, such as branching methods, decomposition methods (column generation, row generation), dynamic programming, heuristics and metaheuristics. None of these papers consider the potential flexibility of relaxing the coupling constraints, which would allow transfers to occur as stated above in Section 1. Two very exhaustive reviews and discussions of the various approaches proposed for solving dial-a-ride problems (DARP), are found in Bodin et al. (1983, 1995). They classify the pickup and delivery problem into static and dynamic cases, with single and multi-vehicle and, with and without time windows. In addition, Solomon and Desrosiers (1988) published an excellent review of the vehicle routing problem with time window constraints. Recently, Toth and Vigo (2002) develop a very complete survey of the methods proposed to solve the PDP until 2001. Berbeglia et al. (2007) have made the latest (to the best of our knowledge) complete updated review, proposing a new classification scheme of pickup and delivery problems. Besides, DARP can include several operational constraints such as time windows, maximum allowable ride time, and other quality service constraints (Cordeau and Laporte, 2003b).

With regard to exact methods, Psaraftis (1980, 1983) and Desrosiers et al. (1986) use dynamic programming techniques to solve the pickup and delivery problem and the pickup and delivery problem with time windows, all of them defining the concept of *state of the system* for each particular problem. Dumas et al. (1991) propose a column generation procedure to solve the pickup and delivery problem with time windows, using the Dantzig–Wolfe decomposition. Their method is able to solve instances of size up to 50 customers. Ruland and Rodin (1997) formulate the pickup and delivery problem as a *mixed-integer programming problem* (mip) and propose four kinds of valid inequalities for the problem, based on the *traveling salesman problem with precedence constraints*. They use the cuts on a branch-and-cut procedure to solve instances up to 15 customers. Lu and Dessouky (2004) solve the pickup and delivery problem with time windows using a branch-and-cut technique, based on new valid inequalities proposed by the authors. They solved instances up to 17 customers. More recently, Dumitrescu (2005) provides several valid inequalities for solving the traveling salesman problem with pickups and deliveries (TSPPD), establishing those among all known inequalities that define facets of the TSPPD polytope.

Although many exact methods have been developed for solving variants of the PDP, none of them actually avoid the complexity of the problem, limiting their solution power to small size problems. This evident drawback motivated the development of good heuristics to solve medium and large scale systems. For example, Sexton and Bodin (1985a,b) propose an heuristic method to solve the pickup and delivery problem, based on Benders decomposition. They use as objective function the customers disutility, decomposing the problem into a routing problem (hard) and a scheduling problem (easy).

Jaw et al. (1986) propose an insertion heuristic to solve the pickup and delivery problem with hard time windows. Madsen et al. (1995) adapt the Jaw et al. (1986) insertion approach to solve a dynamic PDP. Cullen et al. (1981) solve the pickup and delivery problem, introducing the concept of *cluster*, or spatial window, to apply their optimization procedure. A difficulty of this method appears when there are customers whose origin and destination are in different clusters. To deal with this issue, Dumas et al. (1989) propose the incorporation of *mini-clusters*, each one representing a possible route portion. Ioachim et al. (1995) utilizes the Dumas et al. (1991) philosophy to approximate the global routing solution allowing the insertion of additional requests during the day of operation. The approach can solve problems of about 200 customers and 85 miniclusters in a very reasonable computation time.

Other approximation solution procedures for the static case include the column management scheme by Savelsbergh and Sol

(1998), the parallel insertion heuristic developed by Toth and Vigo (1997), and the parallel cooperative multi-search method by Le Bouthillier and Crainic (2005).

Xu et al. (2003) propose a column generation algorithm for the PDP including a series of real-world logistics constraints. Subproblems are solved by an heuristic procedure. The authors are able to find near optimal solutions from random instances of up to 200 requests in a reasonable computation time. Nanry and Barnes (2000) solve instances of 100 customers of the multi-vehicle PDP by using a reactive tabu search method. Lau and Liang (2002) formulate a problem based on a weighted objective function, comprising fleet size as well as travel cost as decision variables. The problem is solved by heuristics.

In the last years, some sophisticated approximated techniques have been used for solving dynamic PDP instances, mainly genetic algorithms (Jih and Yun-jen, 1999; Haghani and Jung, 2005; Osman et al., 2005), and various metaheuristics (Li et al., 2005; Tarantilis et al., 2005; Bianchessi and Righini, 2007; Cordeau and Laporte, 2003a). Cordeau and Laporte (2003a) developed a tabu search heurtistics for the DARP with time windows, where the objective was to minimize routing costs. Moreover, Cordeau (2006) developed a branch and cut algorithm to solve the same type of problem.

Since the objective of our research is to introduce the transfer option into the original PDP scheme, it is also relevant to review the literature regarding transfer operation in the context of integrated systems, combining fixed and reroutable route portions. At this point, we recognize that many authors have introduced the idea of transfer into dial-a-ride systems, but there is no evidence of a strict formulation of such a problem. Instead, most of them add transfer points and propose operational strategies for efficient transfer operations. Aldaihani and Dessouky (2003) developed a system that integrates fixed routes with a general pickup and delivery problem creating what the authors call a hybrid routing problem. Quadrifoglio et al. (2006), develop some bounds on the maximum longitudinal velocity to evaluate the performance of such a hybrid system. Following the same research line, Malucelli et al. (1999) and Crainic et al. (2001) propose and formulate the operation of a new flexible collective transportation system, combining conventional fixed route lines with lines based on flexible itinerary and timetable. Liaw et al. (1996), Hickman and Blume (2000) develop integrated approaches, solved with insertion heuristics. The research by Horn (2002a,b) proposes an interesting insertion heuristic algorithm and is mixed in the sense of combining different types of transportation systems. Mitrović-Minić and Laporte (2006) developed a two-phase heuristics for the PDP with transfers and time windows (first a random multi-start cheapest insertion procedure followed by a request reinsertion phase), concluding (for instances of 100 requests and 4 transfers) that transfers generate significant gains in scenarios where requests are clustered.

Finally, we mention the scheme developed by Cortés and Jayakrishnan (2002, 2003) called High Coverage Point to Point Transit (HCPPT) based on Shuttle-style operations with a large number of deployed vehicles. The system design ensures that no more than one transfer is needed for the travelers, by using transfer hubs as well as reroutable and non-reroutable portions in the vehicle's travel plans. This bound in the maximum allowable number of transfers is imposed in the design in order to somehow ameliorate the extra disutility on passengers due to transfer operations, which is clearly non-linear with respect to the number of transfers performed during a trip.

In the next section, we will generalize the traditional mixed-integer formulation of the static PDP, by incorporating transfers into the operation. This mathematical approach includes special modeling of transfer points as well as additional variables to identify specific clients and their interaction with vehicles at pickup, delivery and transfer points.

3. The pickup and delivery problem with transfers

In this section, we present a mathematical model for the PDPT. We start conceptualizing the object transfer point and the way that it will be modeled. The PDPT formulation is built by systematically adding variables and constraints. At this point, and in order to clarify the modeling approach and the formulation we distinguish two concepts: request and passenger. The former is a mathematical object represented by an origin, a destination and the number of passengers to be picked up (size). The latter is the real object to be transported (it might be people, freight, etc.). Thus, a request may be seen as a set of passengers (e.g. a customer-party) traveling from the same origin to the same destination. In the proposed formulation, we do not allow passengers belonging to the same request to be split into different vehicles.

3.1. Transfer representation and modeling

Let us describe the transfer point as a special object into the modeling context. In classical PDP, three kind of nodes can be distinguished: depots, origins and destinations. At every origin or destination node, we identify only one operation associated with vehicles, which is either loading (origin) or unloading (destination) of passengers, but not both. At depots, actually there is no load/unload of passengers since vehicles start and finish their routes empty.

In the case of transfer nodes, vehicles may either load or unload passengers. To capture the difference between both operations, we propose to split every transfer node r, into two separate nodes, $s(r)$ (start node) and $f(r)$ (finish node), within which, only one operation associated with vehicles load/unload is allowed.

When a vehicle passes through a transfer point r, it first enters node $s(r)$ to allow set down of passengers who wish to transfer to a different vehicle. The vehicle then proceeds (at least notionally) to node $f(r)$, where waiting passengers can get on board. In Fig. 1, we show graphically how vehicles enter and leave the transfer.

This operational modeling scheme at transfer nodes will be very useful to clarify the formulation as presented next (see Fig. 2).

3.2. PDPT formulation

Let us first define the objects that describe a valid PDPT instance (see Table 1). Additionally, let $E = E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5$ be the set of links, with $E_i, i = 1, \ldots, 5$ defined as follows:

$$
\begin{aligned}
E_1 &= M^+ \times (N^+ \cup s(T)) \cup \{(k^+, k^-) : k \in M\},\\
E_2 &= (N^+ \times (N \cup s(T))) \setminus \{(i^+, i^+) : i \in C\},\\
E_3 &= (N^- \times (N \cup s(T) \cup M^-)) \setminus \{(i^-, i^+), (i^-, i^-) : i \in C\},\\
E_4 &= \{(s(r), f(r)) : r \in T\},\\
E_5 &= (f(T) \times (N \cup M^- \cup s(T))) \setminus \{(f(r), s(r)) : r \in T\}.
\end{aligned}
$$

This definition of the set of links considers all possible edges except those not reasonable, like direct paths from a depot to a node $i^- \in N^-$ or from a node $i^+ \in N^+$ to the depot. In order to clarify the formulation, let us consider an illustrative example of a configuration with 2 vehicles, 5 requests and 1 transfer point (if the reader wants to explore the features and data associated with the instance, the associated databases can be downloaded from the corresponding author's website (Contardo, 2008). In the figure, the main circle represents the Central Depot (CD) that plays the role of origin and destination depot associated with both vehicles, and represents also the transfer station (see the zoom of station CD at the right side of

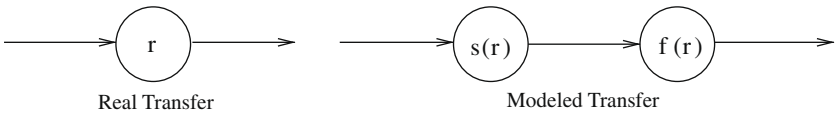

Fig. 1. Transfer representation and modeling.

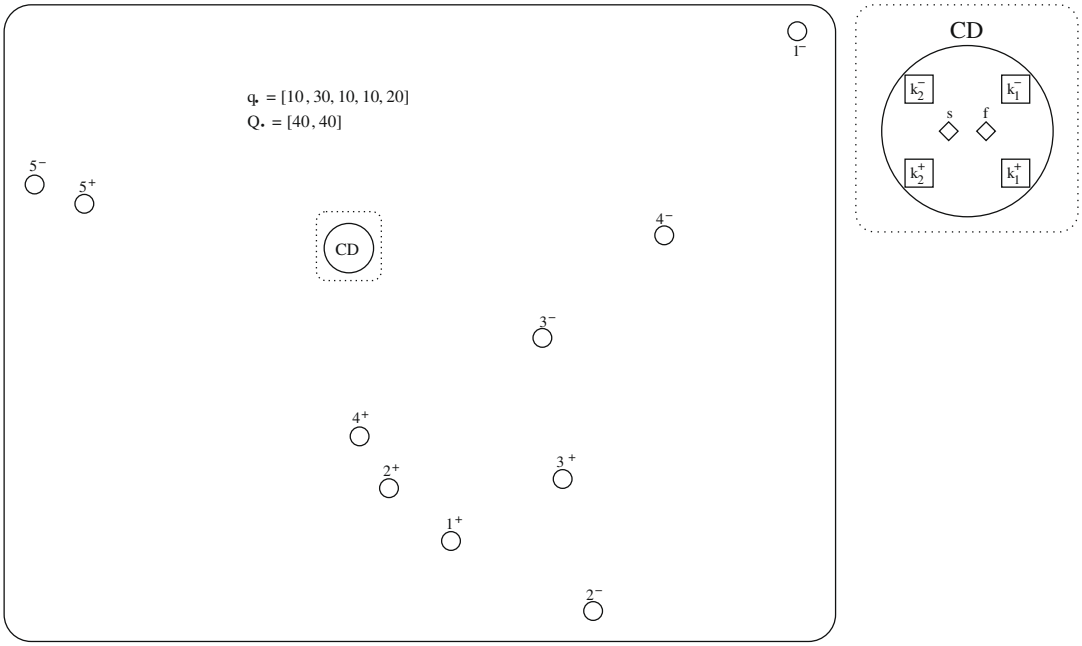

Fig. 2. Example instance for 2 vehicles, 5 requests and 1 transfer.

Table 1
PDPT objects.

Object	Meaning	Comment
M	Set of vehicles	
C	Set of requests	
T	Set of transfer points	
M^+	Set of origin depots for vehicles	$M^+ = \{k^+ : k \in M\}$
M^-	Set of destination depots for vehicles	$M^- = \{k^- : k \in M\}$
N^+	Set of origin nodes for requests	$N^+ = \{i^+ : i \in C\}$
N^-	Set of destination nodes for requests	$N^- = \{i^- : i \in C\}$
N	Set of nodes associated with requests	$N = N^+ \cup N^-$
$s(r)$	Start node of transfer $r \in T$	
$f(r)$	Finish node of transfer $r \in T$	
$s(T)$	Set of start nodes of transfers	$s(T) = \{s(r) : r \in T\}$
$f(T)$	Set of finish nodes of transfers	$f(T) = \{f(r) : r \in T\}$
V	Set of nodes	$V = M^+ \cup M^- \cup N \cup s(T) \cup f(T)$
q_i	Size of request $i \in C$	
Q_k	Capacity of vehicle $k \in K$	
t_{ij}	Minimum ride time from node i to node j	

the figure). All nodes inside the CD are described by the same coordinates.

For each $i \in V$ we define $V^-(i)$ as the set of nodes $j \in V$ such that $(j,i) \in E$. In the same way, we define $V^+(i)$ as the set of nodes $j \in V$ such that $(i,j) \in E$.

In what follows we describe the pickup and delivery problem with transfers (PDPT) in two different ways. First, we conceptually state the properties required to have a feasible solution of such a problem. Second, we relate these properties with a mixed-integer linear problem over the graph $G = (V,E)$. We start defining the concept of *route*.

Definition 1 (*Route*). A route in the graph $G = (V,E)$ is a *simple path* P in G starting at $k^+ \in M^+$ and ending at $k^- \in M^-$.

Then, to conceptually define the PDPT, we realize that for any feasible solution, the following statements must hold

(pdpt1) For every vehicle $k \in M$, the graph induced by the arcs used by vehicle k is exactly a route R_k starting at k^+ and ending at k^- (without cycles).
(pdpt2) Every request must be served, that is, its origin and destination nodes must be visited exactly once.
(pdpt3) For every request, its origin node must be visited before its destination node.
(pdpt4) If a passenger reaches transfer $r \in T$ on vehicle $k_1 \in M$, then he must leave the transfer on a vehicle $k_2 \in M$, such that vehicle k_1 arrives at r before vehicle k_2 leaves r.
(pdpt5) Vehicles do not exceed their capacity.

To model the vehicle routes, we define the binary variable x_{ij}^k for every $(i,j) \in E$, and for every $k \in M$, which is equal to 1 if vehicle k uses link $(i,j) \in E$, and 0 otherwise. The following set of constraints corresponds to the basic route constraints

$$\sum_{i \in V^+(k^+)} x_{k^+ i}^k = 1 \quad \forall k \in M, \tag{1}$$

$$\sum_{i \in V^-(k^-)} x_{ik^-}^k = 1 \quad \forall k \in M, \tag{2}$$

$$\sum_{m \in M^+} \sum_{i \in V^+(m)} x_{mi}^k \leqslant 1 \quad \forall k \in M, \tag{3}$$

$$\sum_{j \in V^+(i)} x_{ij}^k - \sum_{j \in V^-(i)} x_{ji}^k = 0 \quad \forall i \in N,\ \forall k \in M, \tag{4}$$

$$\sum_{i \in V^-(s(r))} x_{is(r)}^k = x_{s(r)f(r)}^k \quad \forall r \in T,\ \forall k \in M, \tag{5}$$

$$\sum_{j \in V^+(f(r))} x_{f(r)j}^k = x_{s(r)f(r)}^k \quad \forall r \in T,\ \forall k \in M. \tag{6}$$

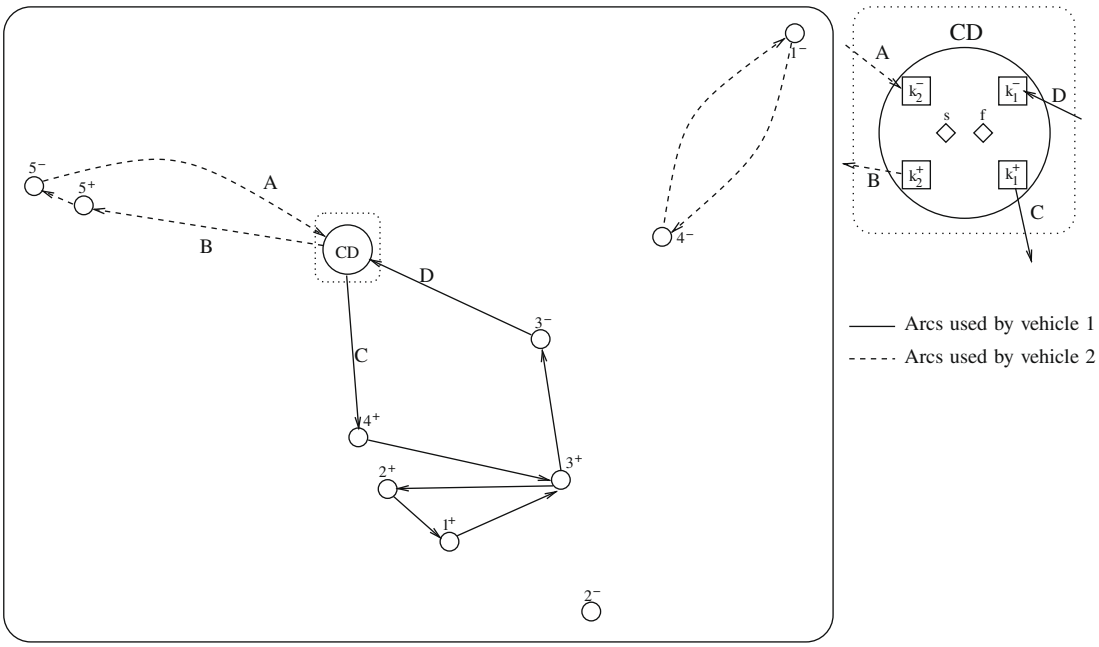

Fig. 3. Example of feasible solution defined by constraints (1)–(6).

Constraints (1) and (2) assure that every vehicle leaves from its origin depot and arrives to its destination depot. Note that from the definition of E, nodes in M^+ are sources of flow, while nodes in M^- are sinks. Constraint (3) assures that a single vehicle cannot start two routes from different origin depots. Constraint (4) is a flow conservation constraint of nodes in N. Constraints (5) and (6) show the flow conservation at transfer nodes. Jointly, these constraints define for every vehicle $k \in M$, a set of routes R_k starting at k^+, ending at k^- and (eventually) containing cycles, as stated by Proposition 18 and illustrated in Fig. 3, which shows how a feasible solution will look like for the example instance presented before.

At this point, the set of constraints (1)–(6) does not assure that property (pdpt2) holds. Actually, at this point none of the conditions (pdpt1) to (pdpt5) hold. Indeed, in order to fulfill (pdpt2), we require to add the following constraints

$$\sum_{k\in M}\sum_{j\in V^-(i^-)} x^k_{ji^-} = 1 \quad \forall i \in C, \tag{7}$$

$$\sum_{k\in M}\sum_{j\in V^+(i^+)} x^k_{i^+j} = 1 \quad \forall i \in C. \tag{8}$$

Constraints (7) force that exactly one vehicle stops at the destination node of request i, while constraints (8) refer similarly to the requests origin node. At this stage, we recognize the first relevant difference with the classic pickup and delivery problem formulation, in which the same vehicle must attend both the origin and the destination nodes of each request. In this new formulation, the origin–destination pair of a single request may be attended by two different vehicles. Proposition 19 assures that the induced graph described by constraints (1)–(8) comprises only routes and simple cycles. In the example, Fig. 4 shows how a feasible solution will look like with these added constraints.

With the aforementioned constraints, conditions (pdpt1), (pdpt3) and (pdpt4) still do not hold. In order to fulfill these relevant conditions, we define the following real variables (associated with arrival time at nodes)

D_i: Time at which node $i \in N$ is attended.
$D^k_{s(r)}$: Time at which vehicle $k \in M$ arrives at transfer $r \in T$.
$D^k_{f(r)}$: Time at which vehicle $k \in M$ leaves transfer $r \in T$.

Then, we add the following constraints to the problem

$$x^k_{k^+i^+} = 1 \Rightarrow t_{k^+i^+} \leqslant D_{i^+} \quad \forall k \in M,\ \forall i \in C, \tag{9}$$

$$x^k_{k^+s(r)} = 1 \Rightarrow t_{k^+s(r)} \leqslant D^k_{s(r)} \quad \forall k \in M,\ \forall i \in C, \tag{10}$$

$$x^k_{ij} = 1 \Rightarrow D_i + t_{ij} \leqslant D_j \quad \forall k \in M,\ \forall (i,j) \in N^2 \cap E, \tag{11}$$

$$x^k_{is(r)} = 1 \Rightarrow D_i + t_{is(r)} \leqslant D^k_{s(r)} \quad \forall k \in M,\ \forall i \in N,\ \forall r \in T, \tag{12}$$

$$x^k_{s(r)f(r)} = 1 \Rightarrow D^k_{s(r)} + t_{s(r)f(r)} \leqslant D^k_{f(r)} \quad \forall k \in M,\ \forall r \in T, \tag{13}$$

$$x^k_{f(r)j} = 1 \Rightarrow D^k_{f(r)} + t_{f(r)j} \leqslant D_j \quad \forall k \in M,\ \forall j \in N,\ \forall r \in T, \tag{14}$$

$$x^k_{f(r)s(u)} = 1 \Rightarrow D^k_{f(r)} + t_{f(r)s(u)} \leqslant D^k_{s(u)} \quad \forall k \in M,\ \forall r \in T,\ \forall u \in T \setminus \{r\}. \tag{15}$$

These constraints eliminate the chance of having a cycle with positive length as part of the solution of the problem. In general, most of the t_{ij}'s may be assumed to be strictly positive, and therefore, we should have very few zero-length cycles. In order to avoid adding subtour elimination constraints for dealing with these zero cost cycles, we can consider a very small travel time between any pair of nodes in the graph, even if they are located in the same physical place. In fact, one could interpret such a travel time as a preparation time for pick up, delivery or transfer, depending on the node under analysis. Aware of this modeling trick, henceforth we will assume that there are no zero-length cycles. Under this assumption, and according to constraints (9)–(15), condition (pdpt1) holds, however conditions (pdpt3)–(pdpt5) do not, as shown in Fig. 5.

This set of constraints can be written as a linear expression in the variables of the problem using the big-M technique (Desrosiers et al., 1995; Desrochers et al., 1988).

We now introduce conditions which involve substantial differences from the standard PDP formulation. In the classical PDP a request's origin and destination nodes always belong to the same route. By contrast, an explicit *tracking* of each request is required in the PDPT, in order to satisfy conditions (pdpt3)–(pdpt5). Let us denote z^{ki}_j a binary variable which is equal to 1 if request $i \in C$ is on vehicle $k \in M$ when it arrives at node $j \in V$, and 0 otherwise, for all $i \in C, k \in M, j \in V$. Then the following constraints involving these new variables must hold

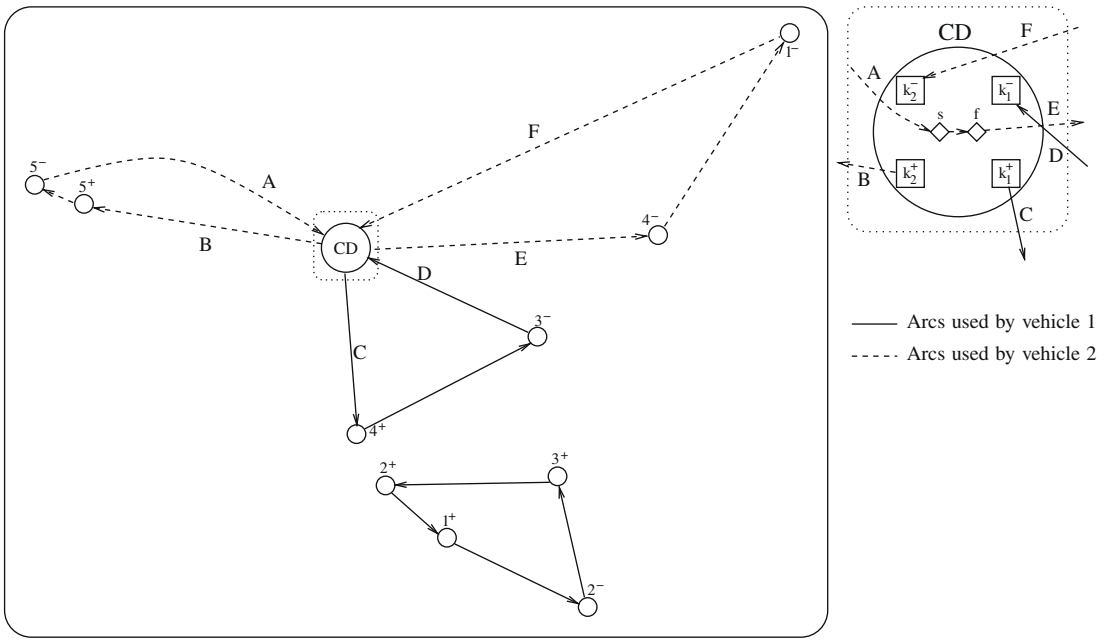

Fig. 4. Example of feasible solution defined by constraints (1)–(8).

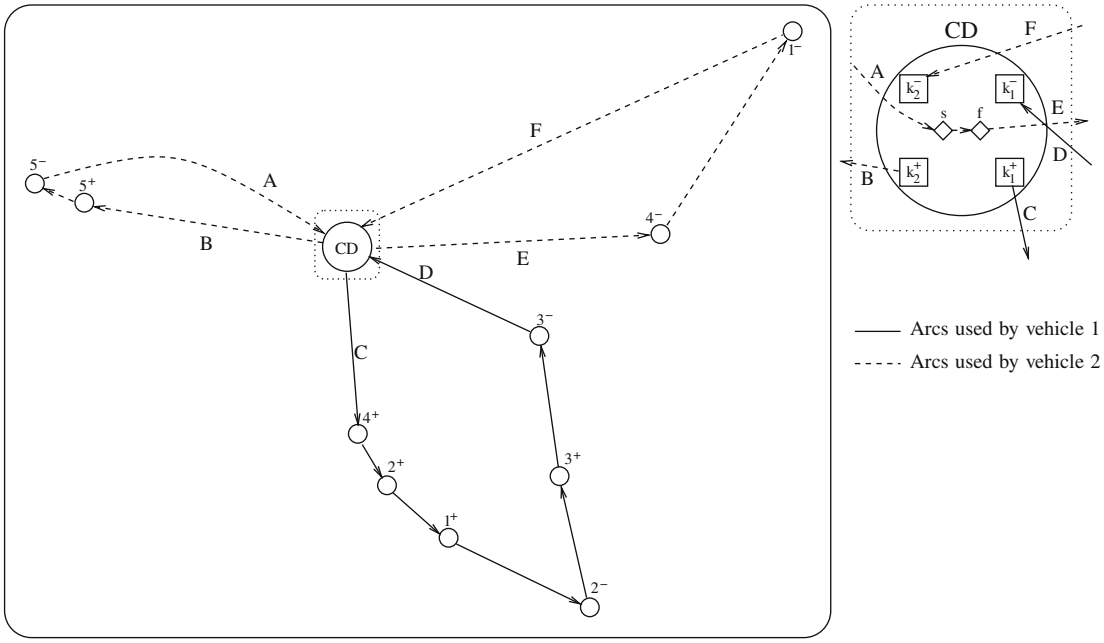

Fig. 5. Example of feasible solution defined by constraints (1)–(15).

$$z^{ki}_{k^+} = z^{ki}_{k^-} = 0 \quad \forall k \in M,\ \forall i \in C, \tag{16}$$

$$x^k_{lj} = 1 \Rightarrow z^{ki}_l = z^{ki}_j \quad \forall k \in M,\ \forall i \in C,\ \forall (l,j) \in E^T \text{ such that } l \neq i^+, i^-, \tag{17}$$

$$x^k_{i^+j} = 1 \Rightarrow z^{ki}_j = 1 \quad \forall k \in M,\ \forall i \in C,\ \forall j \in V^+(i^+), \tag{18}$$

$$x^k_{i^-j} = 1 \Rightarrow z^{ki}_j = 0 \quad \forall k \in M,\ \forall i \in C,\ \forall j \in V^+(i^-) \tag{19}$$

$$\sum_{k\in M} z^{ki}_{s(r)} - \sum_{k\in M} z^{ki}_{f(r)} = 0 \quad \forall r \in T,\ \forall i \in C, \tag{20}$$

$$\sum_{\{l|(l,j)\in E\}} x^k_{lj} = 0 \Rightarrow \sum_{i\in C} z^{ki}_j \leqslant 0 \quad \forall k \in M,\ \forall j \in V \setminus \{k^+, k^-\}, \tag{21}$$

where $E^T = E \setminus \{(s(r), f(r)) | r \in T\}$ is the set of all links except those joining $s(r)$ with $f(r)$, for $r \in T$. These variables keep track of each client while he(she) is traveling from stop to stop. This modeling approach is quite useful when transferring passengers from one vehicle to another at transfer points, unlike traditional PDP formulations that do not need such information to determine optimal routing schemes.

Constraints (16) assure that vehicles start and finish their routes empty at their origin and destination depots. Constraints (17) avoid passengers to get on (off) a vehicle if the node is different from his origin (destination). Constraints (18) (resp. (19)) establish that passengers get on (resp. get off) to (resp. from) the vehicle which reach

his origin (resp. destination) node. Constraints (20) assures that passengers who arrive at a transfer on any vehicle must then leave the transfer on another vehicle (eventually the same one). Finally, constraints (21) establishes that if z_j^{ki} is equal to 1 then vehicle k reaches node j (reciprocally if vehicle k does not get to node j, then z_j^{ki} is equal to 0, for all $i \in C$). These constraints represent the processes of loading and unloading passengers. In the pickup and delivery problem with transfers the variables z along with their associated constraints are fundamental, because when a vehicle arrives at a transfer, we require not only the size of the request, but also the identification of such a request, and that is possible through the set of variables z. As before, these constraints can be written as linear expressions by using again the Big-M technique.

At this point, Condition (pdpt3) holds if all pairs $(origin, destination)$ of requests are in the routes of the same vehicles. But as we know, origin and destination of a request may be in different vehicles, provided that our objective is to relax such a constraint, and visualize conditions for transferring passengers between vehicles.

Although this set of constraints keeps information of how vehicles load and unload passengers, there is no constraint that assures the satisfaction of condition (pdpt4). To deal with this situation, the following constraint must be added

$$z_{s(r)}^{ki} + z_{f(r)}^{vi} = 2 \Rightarrow D_{s(r)}^{k} + \Delta \leqslant D_{f(r)}^{v} \quad \forall r \in T,\ \forall k, v \in M,\ \forall i \in C, \tag{22}$$

where Δ is a positive value that could be interpreted as the time needed for passengers to be ready to get on a vehicle after being dropped at the transfer location by another vehicle. Lemmas 20 and 21 and Proposition 22 prove that condition (pdpt3) holds. Fig. 6 shows a valid solution of the example satisfying constraints (1)–(22). In this example, passengers belonging to requests 1 and 3 transfer from vehicles 1 to 2.

With regard to the vehicle capacity constraint, if Q_k is the capacity of the vehicle k, then the following expression must be added to the formulation

$$\sum_{i \in C} q_i z_j^{ki} \leqslant Q_k \quad \forall k \in M,\ \forall j \in V \tag{23}$$

fulfilling condition (pdpt5).

A big computational issue – not mentioned so far – related to possible solution methods for this optimization problem, is that we added $O(|V||M||C|)$ binary variables not included in the original pickup and delivery problem formulation (which are the z variables). However, we realize that most of these variables can be relaxed to take real values within interval $[0,1]$, not affecting the problem solution, as stated by Proposition 23. This way, the number of binary variables added to the problem, with respect to the original PDP formulation is $O(|T||M||C|)$, less than $O(|E||M|)$, which is the number of binary variables in the original PDP. Note also that if the maximum allowed number of transfers per passenger were one or less, constraints (C.4) would not be required and several binary variables could be removed, speeding up a branch and bound method based on this formulation.

3.3. Time windows inclusion

In most applications in the literature, the PDP problem considers either hard or soft time windows constraints, for pickup, delivery or both. Suppose that nodes must be served within a time window $[l_i, u_i]$ (hard time windows). In terms of the variables of the problem, this may be written just as

$$l_i \leqslant D_i \leqslant u_i \quad \forall i \in N \tag{24}$$

and all feasible solutions of the problem must satisfy this set of constraints.

Another way to include time windows is by penalizing the violation of these constraints (soft time windows). For that, we define the variables δ_i, σ_i and add the following constraints

$$\delta_i \geqslant D_i - u_i \quad \forall i \in N, \tag{25}$$
$$\sigma_i \geqslant l_i - D_i \quad \forall i \in N, \tag{26}$$
$$\delta_i, \sigma_i \geqslant 0 \quad \forall i \in N. \tag{27}$$

The value that variables δ_i, σ_i take, represents the violation of the time windows. Then, a penalty function $p(\delta, \sigma)$ is added into the objective function to capture this effect.

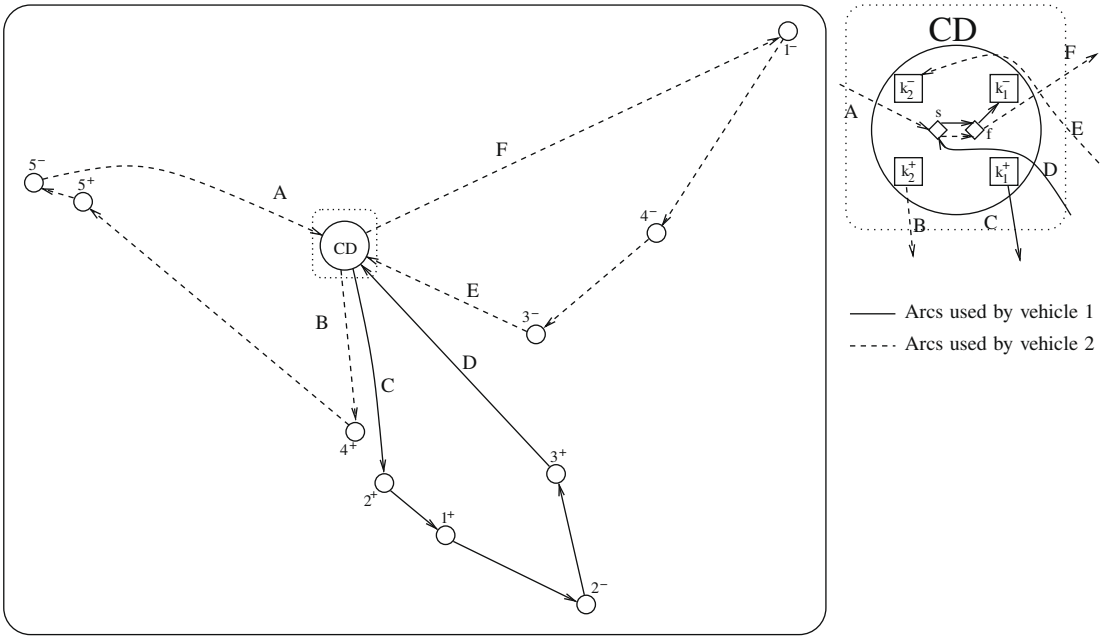

Fig. 6. Example of feasible solution defined by constraints (1)–(22).

3.4. Multiple stops at a transfer

Constraints (5) and (6) not only consider the flow conservation at transfer nodes, but also restrict each vehicle to reach a specific transfer node at most once. In most situations, the maximum number of times a vehicle stops at a transfer r should be part of the decision and not fixed a priori. To include this extra flexibility, the modeler just have to add as many copies of a transfer node as required. For example, if due to physical conditions associated with a valid instance, vehicles cannot reach a transfer more than τ times, it would be enough to add τ copies of every physical transfer, and therefore the instance would have $\tau|T|$ transfer nodes available, leaving the decision of how many times every vehicle passes through a transfer node as part of the optimization procedure. Analytically, let T be the original set of transfers, and let us define, for every transfer $r_j \in T$, the set of copies $T_j = \{r_j^i\}_{i=1}^{\tau}$ as mentioned before. The new transfer set is now $T' = \cup_{r_j \in T} T_j$. Now, we will allow a passenger belonging to request i who arrives to a copy r_j^l of transfer r_j, to leave transfer r_j from any other copy, say $r_j^{l'}$, by replacing the constraint (20) by

$$\sum_{k\in M}\sum_{l=1}^{\tau} z^{ki}_{s(r_j^l)} = \sum_{k\in M}\sum_{l=1}^{\tau} z^{ki}_{f(r_j^l)} \quad \forall i \in C,\ \forall r_j \in T. \tag{28}$$

As a result of this new extra flexibility (passengers may enter to a copy of a transfer and exit from another), we realize that now the order in which a vehicle visits the copies of a transfer r_j is not important. Therefore, and for breaking the evident symmetry induced by this modeling approach, the following constraints may be added

$$x^k_{s(r_j^l)f(r_j^l)} \geqslant x^k_{s(r_j^{l+1})f(r_j^{l+1})} \quad \forall k \in M,\ \forall r_j \in T,\ \forall l = 1,\ldots,\tau-1 \tag{29}$$

forcing the vehicles to visit the copies of a given transfer $r_j \in T$ in order, starting at the first one and ending at the last one. The constraint (23) must also be replaced by

$$z^{ki}_{s(r_j^l)} + z^{vi}_{f(r_j^{l'})} = 2 \Rightarrow D^k_{s(r_j^l)} + \Delta \leqslant D^v_{f(r_j^{l'})} \quad \forall r_j \in T,\ \forall 1 \leqslant l,\ l' \leqslant \tau,\ \forall k, v \in M,\ \forall i \in C. \tag{30}$$

3.5. Time windows at transfers

Sometimes, waiting at a transfer location can be annoying for the passenger. Therefore, it seems very reasonable for the system designer to somehow bound the maximum time a passenger can wait in a transfer node. In the proposed PDPT formulation, this extra feature can be forced by adding an additional constraint as a function of a parameter Γ that can be interpreted as the maximum allowable time a passenger can wait at a transfer location. Analytically,

$$z^{ki}_{s(r)} + z^{vi}_{f(r)} = 2 \Rightarrow D^k_{s(r)} + \Gamma \geqslant D^v_{f(r)} \quad \forall r \in T,\ \forall k, v \in M,\ \forall i \in C. \tag{31}$$

This extra constraint together with constraint (22), generate a time window for the time that passengers wait at transfer points, which corresponds to the interval $[\Delta, \Gamma]$.

3.6. Objective functions for different purposes

Even though we have already formulated the PDPT, it is not clear how to handle the possible objective functions in terms of the problem variables. Next, we present some of the most utilized objective function expressions in the literature:

Common expressions used as objective function

Box 1

$\sum_{k\in M}\sum_{(i,j)\in E} t_{ij} x^k_{ij}$	Total ride time spent by vehicles
$\alpha\sum_{i\in C} D_{i^+} + \beta\sum_{i\in C}(D_{i^-} - D_{i^+})$	Waiting and ride times of requests
$\sum_{k\in M}(1 - x^k_{k^+k^-})$	Fleet size
$f(\delta,\sigma) \geqslant 0$ and increasing by component	Time windows violations

For a planner, an important task is to ensure that the objective function applied in practice encapsulates policies aligned with the mission of the agency operating the transport service. The objective function expressions presented above show the diversity of options the planner has; some expressions are oriented exclusively to offer the best level of service for users, others are focused on the minimization of the operational cost of vehicles instead (directly associated with the total time or distance traveled by them). Within the proposed expressions, we also report a potential fleet size minimization problem subject to offering a certain level of service, which corresponds to a system design problem. That case should be complemented with a demand study in order to quantify the potential customers willing to use the system according to the offered level of service, and based on that, the minimization can provide insights to determine the proper fleet size in operation.

From a system standpoint (operator and users), the proper definition of an objective function should combine several of the above proposed objective function terms, including both operator and user costs (the latter quantified by the level of service measured through travel time, waiting time, time windows violation, and so on) These two dimensions, on the one hand the operator who pursue the minimization of operational costs, and, on the other hand the users who claim for a good level of service, are opposite objectives. In fact, offering a better service level implies more direct trips, this means, lower vehicle occupation and consequently, higher operation costs to respond to the same demand with the same fleet. To the opposite, more efficient routing policies from an operation standpoint necessarily result in higher occupation rates, longer routes, and consequently, worse level of service (higher travel and waiting times for users, long time windows violation, etc.). In practical terms, some policy rules can be established to make reasonable planning and fleet dispatch decisions. For example, the operator may guarantee a minimum level of service (e.g. an upper limit on waiting time and travel time), and then simply define the objective function as minimization of operating costs, as in Horn (2002b). A multiobjective optimization framework can be used to properly consider the different objective function terms and their importance depending on the specific conditions of real-world implementations for further analysis and routing policies design.

4. PDPT model validation

What we have developed in this paper is a strict formulation of a static pickup and delivery problem with transfers, assuming that the transfer positions are fixed and known in advance. The question then arises whether the new formulation really does permit more efficient transport solutions than the PDP. For example, adding a transfer point in an optimal PDP route (i.e. a generated as part of a solution to the PDP) would increase the total ride time; we consider now whether this additional cost can be offset by the flexibility afforded by the provision of transfer points, such as to

improve the overall efficiency of the system. To visualize that a positive trade off is possible, we have considered the same instance utilized before in Section 3.2 to illustrate the construction of a feasible PDPT solution. In fact, suppose that we want to minimize (as objective function) the total ride time, namely $\sum_{k\in M,(i,j)\in E} t_{ij}x_{ij}^k$. The t_{ij}'s are calculated as the Euclidean distance between two points plus a service time. The instance data can be found in the corresponding author's website Contardo (2008). The PDPT solution provided in Fig. 6 is optimal (objective function value equals to 1187). The arcs used as well as the computed ride times are the following

$$\text{Route } 1 = k_1^+ \overset{20}{\to} 2^+ \overset{98}{\to} 1^+ \overset{104}{\to} 2^- \overset{103}{\to} 3^+ \overset{126}{\to} s \overset{1}{\to} f \overset{51}{\to} 1^- \overset{113}{\to} 4^- \overset{105}{\to} 3^- \overset{110}{\to} k_1^-,$$

$$\text{Route } 2 = k_2^+ \overset{15}{\to} 4^+ \overset{124}{\to} 5^+ \overset{96}{\to} 5^- \overset{120}{\to} s \overset{1}{\to} f \overset{0}{\to} k_2^-.$$

If we do not consider transfer operations, the optimal solution under this scheme is the following

$$\text{Route } 1 = k_1^+ \overset{15}{\to} 4^+ \overset{104}{\to} 1^+ \overset{138}{\to} 5^+ \overset{96}{\to} 5^- \overset{166}{\to} 1^- \overset{113}{\to} 4^- \overset{122}{\to} k_1^-,$$

$$\text{Route } 2 = k_2^+ \overset{20}{\to} 2^+ \overset{113}{\to} 2^- \overset{103}{\to} 3^+ \overset{104}{\to} 3^- \overset{110}{\to} k_2^-,$$

whose objective function value is equal to 1204, strictly greater than 1187. It is clear that both solutions are feasible under the corresponding operational schemes. Moreover, the optimality can be proven by using the exact algorithm presented in the following section, forcing the number of transfer operations to 0 in the latter case. It follows that the solution that includes transfers operations is better than the one that avoids them.

5. A branch-and-cut solution method

In this section, we present a branch-and-cut method to solve the PDPT using Benders Decomposition (Benders, 1962) that applies the Combinatorial Benders Cuts introduced by Codato and Fischetti (2004). In this method, the set of constraints is decomposed into pure integer and mixed constraints, and then a branch-and-cut procedure is applied to the resulting pure integer problem, by using real variables and constraints related as cut generators. The key on the success of this method is that those constraints defined by a logical sentence $p \Rightarrow q$ are not modeled using the big-M technique, as usual in a branch-and-bound methodology behind the original PDP formulation. Although this method may be applied only when the objective function is either pure real or pure integer, we only describe the pure integer case, which could be successfully implemented by the authors (for details and results of the pure real objective function implementation, refer to Contardo, 2005). The expression to be minimized is in our case $\sum_{k\in M}\sum_{(i,j)\in E} t_{ij}x_{ij}^k$, representing the total ride time spent by the fleet of vehicles. We must also be able to ensure that the variables and constraints defined by (C.4) are not needed. This may be done by fixing the cardinality of the transfer set equal to 1, or by adding the constraint $\sum_{r\in T}\sum_{k\in M} z_{s(r)}^{ki} \leqslant 1$, which bounds the number of times a passenger may arrive to a transfer to a maximum of 1. In dial-a-ride systems, this last constraint is, in general, required (see Cortés, 2003).

5.1. Solution method details

Consider the problem

$$\min_{x,y} \quad c^t x \tag{32}$$

$$\text{s.to} \quad x \in X \subseteq \{0,1\}^n \tag{33}$$

$$x_{j(i)} = 0 \Rightarrow a_i^t y \leqslant b_i \quad \forall i \in I \tag{34}$$

$$Ey \leqslant f \tag{35}$$

$$y \geqslant 0 \tag{36}$$

which may be written in a fully equivalent manner as

$$\min_{x\in X}\{c^t x + \min\{0^t y : x_{j(i)} = 0 \Rightarrow a_i^t y \leqslant b_i \forall i \in I, Ey \leqslant f, y \geqslant 0\}\}. \tag{37}$$

The previous problem is equivalent to solve the *master problem* (M) defined as

$$\min_{x} \quad c^t x \tag{38}$$

$$\text{s.to} \quad x \in X \subseteq \{0,1\}^n \tag{39}$$

and at every node of the branching tree, say where $\bar{x}$ is known, a *feasibility problem* $(F_{\bar{x}})$ has to be solved defined as

$$\min_{y} \quad 0^t y \tag{40}$$

$$\text{s.to} \quad a_i^t y \leqslant b_i \quad \forall i \in I_{\bar{x}}, \tag{41}$$

$$Ey \leqslant f, \tag{42}$$

$$y \geqslant 0, \tag{43}$$

where $I_{\bar{x}} = \{i \in I : \bar{x}_{j(i)} = 0\}$. The dual of this problem $(DF_{\bar{x}})$ is as follows:

$$\min_{\lambda,\mu} \quad z(\lambda,\mu) = \sum_{i\in I_{\bar{x}}} \lambda_i b_i + \mu^t f \tag{44}$$

$$\text{s.to} \quad \sum_{i\in I_{\bar{x}}} \lambda_i a_i + E^t \mu \geqslant 0, \tag{45}$$

$$\lambda, \mu \geqslant 0. \tag{46}$$

Note that this last problem has the particularity of being always feasible. As a consequence, an infeasibility of $(F_{\bar{x}})$ is equivalent to the unboundness of $(DF_{\bar{x}})$. Moreover, if one of the problems is feasible, the other is feasible too, and the optimal value is 0 for both problems. Note also that if $(DF_{\bar{x}})$ is unbounded, and (λ^*, μ^*) is a feasible solution such that $z(\lambda^*, \mu^*) = -1$ then $\eta(\lambda^*, \mu^*)$ is an unboundness ray, with $\eta > 0$. Then, we just need to study the reduced problem $(S_{\bar{x}})$, instead of $(DF_{\bar{x}})$. The reduced problem is defined as

$$\min_{\lambda,\mu} \quad g^t \lambda + h^t \mu \tag{47}$$

$$\text{s.to} \quad \sum_{i\in I_{\bar{x}}} \lambda_i b_i + \mu^t f = -1 \tag{48}$$

$$\sum_{i\in I_{\bar{x}}} \lambda_i a_i + E^t \mu \geqslant 0 \tag{49}$$

$$\lambda, \mu \geqslant 0 \tag{50}$$

where $g > 0, h > 0$ are weights defined arbitrarily. Thus, the following proposition is the key for defining the cuts

Proposition 2. *If $(S_{\bar{x}})$ is feasible with solution (λ^*, μ^*), then the constraint*

$$\sum_{i\in I_{\bar{x}}:\lambda_i^*>0} x_{j(i)} \geqslant 1 \tag{CBC}$$

holds for the problem (32)–(36). If $(S_{\bar{x}})$ is unfeasible, then $(F_{\bar{x}})$ is feasible.

Proof. Suppose first that $(S_{\bar{x}})$ is unfeasible, which implies that for every (λ, μ) such that (49) holds, then (48) does not. This implies that $(DF_{\bar{x}})$ is bounded, and consequently by duality, $(F_{\bar{x}})$ is feasible. Suppose now that the constraint (CBC) does not hold for (32)–(36), which implies that exists $\bar{y} \geqslant 0$ such that $\bar{x}_{j(i)} = 0$ for all $i \in I_{\bar{x}}, \lambda_i^* > 0$ is feasible. Then, for those $\bar{x}, \bar{y}$, the following holds

$$a_i^t \bar{y} \leqslant b_i \quad \forall i \in I_{\bar{x}} \quad \text{such that } \lambda_i^* > 0. \tag{51}$$

Multiplying by λ_i^* at both sides of this equation and then summing, we obtain

$$\sum_{i\in I_{\bar{x}}:\lambda_i^*>0} \lambda_i^* a_i^t \bar{y} \leqslant \sum_{i\in I_{\bar{x}}:\lambda_i^*>0} \lambda_i^* b_i. \tag{52}$$

Additionally, we have $E\bar{y} \leqslant f$. If we multiply by μ^{*t} the inequality remains, obtaining $\mu^{*t}E\bar{y} \leqslant \mu^{*t}f$. By summing this inequality with (52), we obtain that

$$\sum_{i\in I_{\bar{x}}:\lambda_i^*>0} \lambda_i^* a_i^t \bar{y} + \mu^{*t}E\bar{y} \leqslant \sum_{i\in I_{\bar{x}}:\lambda_i^*>0} \lambda_i^* b_i + \mu^{*t}f, \tag{53}$$

and since $\bar{y} \geqslant 0$ then constraints (48) and (49) cannot both hold simultaneously. □

5.2. Method implementation

As discussed above, for implementing the method we just need to specify the problems used as *master problem* (M) and *feasibility problem* ($F_{\bar{x}}$). As master problem, we used the following

$$\min_{x} \quad \sum_{k\in M}\sum_{(i,j)\in E} t_{ij}x_{ij}^k \tag{54}$$

$$\text{s.to} \quad x \text{ satisfies } (1)-(8), \tag{55}$$

$$x \in \{0,1\}^{|E||M|}, \tag{56}$$

and as feasibility problem, the one obtained after removing the constraints not activated by the logical sentences. Analytically

$$\min_{D,z} \quad 0^t D + 0^t z \tag{57}$$

$$\text{s.to} \quad \bar{x}, D, z \text{ satisfy } (9)-(23), \tag{58}$$

$$D \geqslant 0, \tag{59}$$

$$z \in [0,1]^{|M||C||V|}. \tag{60}$$

5.3. Numerical results

All numerical tests were performed on an AMD Barton 2900+ (2.0 GHz), with 512 MB of RAM PC3200, Windows XP, CPLEX 9.0 and Visual C++ 6. At every instance, customer nodes are located randomly over a city of 1000×1000 distance units^2 (DU^2); vehicles depots are located randomly too; the transfer node is located in the mass center of the customer nodes N. The objective function utilized was the minimization of the total ride time spent by vehicles $\sum_{k\in M}\sum_{(i,j)\in E} t_{ij}x_{ij}^k$.

For testing the speed of the Combinatorial Benders Cuts method, we run it against the pure branch-and-bound method (with default settings), both implemented in CPLEX/C++ using the Concert Technology interface. Instead of running $S_{\bar{x}}$ at every node of the branching tree, we just do that at the integer nodes, which ended up being the fastest implementation of the method. The average speed (in seconds) of both methods is shown in Table 2. The notation "r" means number of requests, "v" denotes the number of vehicles and "t" stands for the number of transfer points. The columns B&B and CBC mean "branch-and-bound" and "combinatorial benders cuts", respectively.

In Table 2 we can appreciate the advantages in terms of running time of the proposed solution algorithm when compared against a traditional branch-and-bound methodology, obtaining savings of about 90%, that should exponentially increase with the problem size.

Table 2
Average CPU time [seconds].

	B&B	CBC
3r 2v 1t	0.385	0.479
4r 2v 1t	8.879	4.073
5r 2v 1t	233.751	17.469
6r 2v 1t	1872.205	119.531

6. Bounding the branching tree

Some key issues of this formulation and the presented method that have to be taken into account are, first, the evident symmetries that the problem presents if vehicles are indistinguishable and the depot is the same for all vehicles, and second, the nature of the variables generate a three-index formulation, that can become very poor for most of the relaxations. Therefore, finding a good way to tight the formulation is crucial. A possible way to deal with these issues is to add some extra constraints into the formulation, which would tight the relaxation. In the following subsections, we propose two family of inequalities that help tight the formulation.

6.1. Eliminating symmetries

Unfortunately, this formulation does not take into account all the possible symmetries of the problem. Given a solution S, let us denote $R(S)$ the set of routes followed by the vehicles and S_k the route followed by specific vehicle k. Provided two solutions of the problem A and B, we say that these solutions are symmetric if there exists a bijection $f : R(A) \to R(B)$ such that for each $k \in M, f(A_k) = B_k$. In other words, these solutions are identical even though we change the numbering of the vehicles.

To eliminate some symmetric solutions we propose the following: Suppose that requests in C are numbered from 1 to $|C|$ and vehicles from 1 to $|M|$. The following set of constraints imply that nodes in C^+ are visited in order by each one of the vehicles.

$$\sum_{l>i}\sum_{j\in V^+(i^+)} x_{i^+j}^l = 0 \qquad \forall i = 1,\ldots,|C|. \tag{61}$$

6.2. Adding redundant constraints

Although the presented set of constraints defines completely the PDPT polytope, sometimes it is recommended to add some extra constraints reflecting specific properties of the modeled system. Then, although they neither increase nor reduce the solution space, they produce a tighter relaxation of the problem, helping reduce the size of the branching tree.

For example, consider a vehicle that serves node i^+. Then this vehicle must either stop at node i^- or pass through a transfer. This constraint may be written as

$$\sum_{j\in V^+(i^+)} x_{i^+j}^k = 1 \Rightarrow \sum_{j\in V^-(i^-)} x_{ji^-}^k + \sum_{r\in T} x_{s(r)f(r)}^k \geqslant 1 \quad \forall k \in M,\ \forall i \in C. \tag{62}$$

7. Conclusions and further research

In this paper, a strict formulation of a generalization of the classic pickup and delivery problem formulation is presented, which adds the extra flexibility of providing the option for passengers to transfer at specific locations (transfer points). In this paper, we formally present the basic arc-based formulation of the problem, for a fixed number and location of transfer stations. We have to mention that the original static PDP formulations were written as arc-based mixed-integer problems by Desrochers et al. (1988), Desrosiers et al. (1995), Savelsbergh and Sol (1995) to understand the basic structure of the mathematical problem. In the case of transfers, a first major contribution of the research line summarized in this paper is to formalize the modified PDP with transfers (PDPT) by writing a generalized arc-based formulation to represent such a system likewise.

In terms of complexity, this formulation adds $O|T||V||M|)$ binary variables in the case where passengers are allowed to make multiple transfers, which against the $O(|V|^2|M|)$ of the classic PDP is quite small. Moreover, in the case $|T| = 0$, we not only obtain the classic PDP solution, but also the complexity in terms of binary variables keeps unchanged.

In addition, the proposed formulation collects more information with regard to customers than that of traditional PDP applications, since z variables keep track of the exact position of every passenger at every stop through vehicle routes, and since we showed they could be relaxed to be real variables, the problem complexity does not change much with respect to the PDP case. One clear advantage of having these extra variables is the capacity of proposing novel objective functions customer-specific depending upon z variables.

We have shown through an example that the PDPT can yield solutions that in broad terms are more efficient than those obtained from the PDP. This result is highly suggestive for further research directed at the task of defining conditions under which a PDPT solution can outperform a PDP solution. Our conjecture is that, under high demand conditions, transfer operations become more and more profitable. As our algorithm can handle only small instances, the proof of this conjecture is a matter of further research. At this point, we believe that the type of algorithms to be used for handling high demand instances should focus on metaheuristics, such as Tabu Search or Evolutionary Algorithms, just to name some options. Further developments on this line of research have a considerable bearing on the strategic design and planning of demand-responsive passenger transport systems.

Additionally, we present an exact solution method, which relies on a branch-and-cut technique based on Benders decomposition algorithm. We have shown that the technique is very effective. As shown in Table 2, the method can reduce CPU by 90%, at least for the instances we tested here. For bigger instances the method explodes, but as the tendency indicates, the computational effort decreases exponentially with respect to a Branch-and-bound. We are currently studying new cuts to be added to the master problem that could greatly improve the performance of the method. Notice that in Section 6 we provide some guidelines for further improvement of the proposed exact method by adding redundant constraints and eliminating symmetries as well. We guess that a polyhedral study will result in even faster solutions, and this is another topic of further research.

Similar to the PDP case without transfers (Dumas et al., 1991), from the formulation we developed here a set-partitioning problem can be written (route-based) in order to develop more efficient algorithms of the column generation type (branch-and-bound, branch-and-price). We are currently working on formalizing a set partitioning problem for the case of a fixed single transfer point. The idea is to split the routes (columns) in three different types: from a pickup point to a transfer point, from a transfer point to a delivery location, and from a pickup to a delivery without any transfer. Binary variables are used to check feasibility of a sequence of routes to accomplish some trips. Bounds on the arrival and departure times at transfer points for each request are obtained from the subproblem. An efficient implementation of this column generation framework will allow us to find exact optimal solutions for real-size problems, where it will be reasonable to search the advantages of allowing transfers in some dial-a-ride operation schemes under specific system configuration, in terms of demand, vehicle supply and transfer point locations.

Acknowledgements

The authors would like to thank Professor Michel Gendreau as well as an anonymous referee for the collaboration with their useful comments and advices. They would also like to thank Project FONDECYT 1030700, Millenium Institute in Complex Engineering Systems and Millenium Nucleus in Information and Randomness by their financial support.

Appendix A. Graph theory definitions

Definition 3 (*Graph*). A graph is a pair $G = (V, E)$ such that $E \subseteq V^2$. To avoid ambiguities we will always suppose that $E \cap V = \emptyset$. We say that V is the set of nodes of G, and it is written $V(G)$ or simply V when there are no possible confusions. Similarly, we say that E is the set of arcs of G and it is written $E(G)$, or simply E when there are no possible confusions.

Definition 4 (*Inverse graph*). Let $G = (V, E)$ a graph. The Inverse Graph of G is defined as the graph $G' = (V, E')$, where $E' = \{(w, v) : (v, w) \in E\}$. We denote the inverse graph of G as $\mathcal{I}(G)$.

Definition 5 (*Subgraph*). Let $G = (V, E)$ be a graph. We will say that a graph $G' = (V', E')$ is a subgraph of G (and we will denote $G' \subseteq G$) if $V' \subseteq V$ and $E' \subseteq E$.

Definition 6 (*Induced graph*). Let $G = (V, E)$ be a graph. Let $V' \subseteq V$, $E' \subseteq E$. We define $G[V']$ as the graph whose vertex are in V' and whose arcs are every $(u, v) \in E$ such that $u, v \in V'$. Similarly, we define $G[E']$ as the graph whose vertex are all $u \in V$ such that $(u, v) \in E'$ or $(v, u) \in E'$ for some $v \in V$, and whose arcs are exactly E'.

Definition 7 (*Neighbors of a node*). Let $G = (V, E)$ be a graph, and let $v \in V$. We denote $\mathcal{N}_G^-(v) = \{w \in V : (w, v) \in E\}$ and we call *nodes entering to* v. Similarly, we denote $\mathcal{N}_G^+(v) = \{w \in V : (v, w) \in E\}$ and we call it *nodes leaving from* v.

Definition 8 (*Degree of a vertex*). Let $G = (V, E)$ be a graph, and let $v \in V$. We denote $d_G^+(v) = |\mathcal{N}_G^+(v)|, d_G^-(v) = |\mathcal{N}_G^-(v)|$ and they are called exit degree of v and entry degree of v, respectively.

Definition 9 (*Path*). Let $G = (V, E)$ be a graph. Let $v_1, \ldots, v_n \in V$. We say that the tuple $P = (v_i)_{i=1}^n$ is a path in G if $\forall i \in \{1, \ldots, n-1\}(v_i, v_{i+1}) \in E$ and all pairs (v_i, v_{i+1}) are different.

Remark 10. Note that if $P = (v_1, \ldots, v_n)$ is a path in a graph G, then the tuple $G(P) = (V(P), E(P))$ with $V(P) = \{v_i : i = 1, \ldots, n\}, E(P) = \{(v_i, v_{i+1}) : i = 1, \ldots, n-1\}$ is a subgraph of G. Sometimes we will replace the notation $G(P)$ just with P.

Remark 11. Note that in a path $P = (v_1, \ldots, v_n)$ always is satisfied

$$d_P^+(v_1) - d_P^-(v_1) = d_P^-(v_n) - d_P^+(v_n) = \begin{cases} 0 & si\ v_1 = v_n, \\ 1 & si\ v_1 \neq v_n. \end{cases} \quad (A.1)$$

Definition 12 (*Simple path*). Let $G = (V, E)$ be a graph. We say that a path $P = (v_1, \ldots, v_n)$ in G is simple if all nodes v_i are different.

Definition 13 (*Open(Closed) path*). Let $G = (V, E)$ be a graph. We say that a path $P = (v_1, \ldots, v_n)$ in G is open(closed) if $v_1 \neq v_n (v_1 = v_n)$.

Definition 14 (*Maximal path*). Let $G = (V, E)$ be a graph. We say that a path $P = (v_1, \ldots, v_n)$ is *maximal* if there is not way of extending it from any of its ending nodes provided that the new sequence is still a path.

Appendix B. Graph theory results

Proposition 15. *Let $G=(V,E)$ be a graph, with $V=M^+\cup N\cup M^-$ such that*

$$d_G^-(v)=0 \quad d_G^+(v)>0 \quad \forall v\in M^+, \tag{B.1}$$

$$d_G^-(v)>0 \quad d_G^+(v)=0 \quad \forall v\in M^-, \tag{B.2}$$

$$d_G^+(v)=d_G^-(v) \quad \forall v\in N. \tag{B.3}$$

Then, every open maximal path $P=(v_1,\ldots,v_n)$ in G of length $\geqslant 2$ is such that $v_1\in M^+, v_n\in M^-$. In particular, the property is valid for every simple maximal path.

Proof. Let $P=(v_1,\ldots,v_n)$ a maximal path open of length $\geqslant 2$ in G, *i.e.* such that $v_1\neq v_n$ and P cannot be extended beyond any of its endings by arcs which are not in P.

Let us see that $v_1\in M^+$. Suppose that $v_1\notin M^+$. Note in the first place that $v_1\notin M^-$, because in such a case, as P is of length at least 2, we would have that $(v_1,v_2)\in E(G)$ which is impossible because $d_G^-(v_1)=0$. If $v_1\in N$, then $d_G^+(v_1)=d_G^-(v_1)$ by hypothesis. But, by the previous rem we have that in an open path $d_P^-(v_1)-d_P^+(v_1)=1$ and in consequence $\exists v_0\in\mathscr{N}_G^-(v_1)$ such that $(v_0,v_1)\notin E(P)$, and by adding v_0 to the left side of P the new sequence would be still a path, contradicting the maximality of P.

To see that $v_n\in M^-$ note first that $\mathscr{I}(G)$ satisfies the hypotheses of the proposition, defining $M^+_{\mathscr{I}(G)}=M^-, M^-_{\mathscr{I}(G)}=M^+$, and, second, that a path P is maximal and open in G *iif.* $\mathscr{I}(P)$ is maximal and open in $\mathscr{I}(G)$. □

Proposition 16. *Let $G=(V,E)$ be a graph satisfying B.1,B.2,B.3. Then $\forall v\in M^+,\exists P_v$ simple path starting at v and ending at M^-. Similarly, $\forall w\in M^-,\exists P_w$ simple path starting at M^+ and ending at w*

Proof. Let us prove that for every $v\in M^+,\exists P_v$ simple path starting at v and ending at M^-. The reciprocal may be obtained from $\mathscr{I}(G)$ just as in the previous proof.

Let $v\in M^+$. As $d_G^+(v)>0$, exists $v_2\in V(G)$ such that $(v,v_2)\in E$. Let Q be a maximal path to (v,v_2). Such path necessarily is open because $d_G^-(v)=0$, and therefore by the Proposition (15) Q ends at M^-. To be more precise, say that $Q=(v_1,\ldots,v_n)$. By Proposition (15), there exists a simple path which connects v_1 y v_n, and by construction, such a path is maximal and simple. □

Proposition 17. *Let $G=(V,E)$ be a graph, with $V=M^+\cup N\cup M^-$, and satisfying B.1 and B.2 and*

$$d_G^+(v)=d_G^-(v)=1 \quad \forall v\in N. \tag{B.4}$$

Then

1. *Every path from M^+ to M^- is simple.*
2. *Every node which is not in any path from M^+ to M^- belongs to a simple cycle.*

Proof. Let us see first that every path from M^+ to M^- is simple. Let $P=(v_1,\ldots,v_n)$ be a path from M^+ to M^-, and suppose that $v_l=v_j=v$ for $2\leqslant l,j\leqslant n-1, l\neq j$. Then (v_{l-1},v), (v,v_{l+1}), $(v_{j-1},v),(v,v_{j+1})\in E(G)$ and therefore $d_G^+(v)=d_G^-(v)\geqslant 2$. Then, by (B.1)–(B.3) we have $v\in N$, which is impossible by (B.4).

Let us see now that every $v\in V(G)$ which dos not belong to any path from M^+ to M^- belongs to some simple cycle. Let $\mathscr{P}=\cup\{G(P): P \text{ is a path from } M^+ \text{ to } M^-\}$ be the graph of all paths from M^+ to M^-. By Proposition (16) the graph $G-\mathscr{P}$ does not contain any node in M^+ or in M^-, and all its nodes are such that $d^+_{G-\mathscr{P}}=d^-_{G-\mathscr{P}}=1$. By *Euler's Theorem* for directed graphs, we conclude that the connected components of $G-\mathscr{P}$ are simple cycles. □

Appendix C. PDPT results

Proposition 18. *Let $(x_{ij}^k)_{(i,j)\in E}^{k\in M}$ satisfy the constraints (1)–(6). Let $E^k=\{(i,j)\in E: x_{ij}^k=1\}$ be the set of arcs used by vehicle k, and $G^k=G[E^k]$ denote the graph induced by those arcs. Then G^k contains a route starting at k^+ and ending at k^-.*

Proof. For notational convenience we define $V^k:=V(G^k)$. Let $M^{k+}=M^+\cap V^k, M^{k-}=M^-\cap V^k, N^k=V^k\setminus(M^+\cup M^-)$. We will prove that for these sets, properties (B.1)–(B.3) hold (see appendix). Thus, by using Proposition 15 we conclude the result.

Let $v\in M^{k+}$, i.e. $d^+_{G^k}(v)>0$ or $d^-_{G^k}(v)>0$, and as $d^-_G(v)=0$ we have (B.1). Let $w\in M^{k-}$, *i.e.* $d^+_{G^k}(w)>0$ or $d^-_{G^k}(w)>0$, and as $d^+_G(w)=0$ we have (B.2). The property (B.3) holds because of the flow conservation at nodes in $N\cup s(T)\cup f(T)$ given by constraints (4)–(6). Constraints (1) and (2) imply that $k^+\in M^{k+}, k^-\in M^{k-}$. Then the proposition 16 assures that exists (at least) one route from k^+ to M^{k-} and one from M^{k+} to k^-. If there were not a route from k^+ to k^- in graph G^k, then we would conclude that the route from M^{k+} to k^- should start from another node $k'^+\neq k^+$, which would violate constraint (3). □

Proposition 19. *Let $x_{ij}^k)_{(i,j)\in E}^{k\in M}$ satisfy constraints (1)–(8). Let $k\in M$. Then the graph G^k (defined as in the previous proposition) is composed by a route from k^+ to k^- plus simple cycles.*

Proof. In fact, we know from the previous proposition that properties (B.1) and (B.2) hold. From constraints (7) and (8) we have that $d^+_{G^k}(v)=d^-_{G^k}(v)=1\forall v\in N^k$, implying that property (B.4) holds. Then, by Proposition 17 we have that (1) every path from M^{k+} to M^{k-} is simple, (2) every node not in any path from M^{k+} to M^{k-} belongs to a simple cycle. Consider all (simple) paths which go from M^{k+} to M^{k-}. By constraint (3) all those paths start in k^+ and, moreover, they start with the same arc (k^+,v). Let us see that all those paths are in fact the same and, as consequence of the previous preposition, that is the path from k^+ to k^-. Let Q be any path from M^{k+} to M^{k-}. Constraints (1) and (3) establish that the new path must start necessarily from the same arc as that of the other path. But now, the ending node of such arc (say v) belongs to i) M^{k-}, and in consequence they are the same path, or ii) $N\cup s(T)\cup f(T)$, then the external degree is in consequence $d^+_{G^k}(v)=1$, and this new path also must use the same second arc as that of the other path. Applying this argument a finite number of times, we can conclude that both paths share all arcs, *i.e.* they are the same path. □

Lemma 20. *Let x,D,z satisfying (1)–(22), and let $i\in C$ be any customer. Let us define the graph G^i as the graph induced by the following set of arcs E^i*

1. *if $j\notin f(T), z_j^{ki}=1$ and $x_{ij}^k=1$ then $(l,j)\in E^i$*
2. *if $j=f(r)\in f(T), z_{s(r)}^{ki}=1$ then $(s(r),f(r))\in E^i$*

Then the graph G^i contains no cycles.

Proof. Note that in case that the graph G^i contains a cycle, exists a sequence $(k_j,r_j,k_{j+1})_{j=1}^J$, with $J\geqslant 2$, satisfying

$$z_{s(r_j)}^{k_j i}=z_{f(r_j)}^{k_{j+1} i}=1 \quad \forall j=1,\ldots,J, \tag{C.1}$$

$$r_1=r_J, \tag{C.2}$$

and for each j there are no transfers between the nodes $f(r_j)$ and $s(r_{j+1})$. In such way, if G^i contains a cycle, by constraints (9)–(15)

and (22) we would have $D^{k_j}_{s(r_j)} < D^{k_{j+1}}_{f(r_j)} < D^{k_{j+1}}_{s(r_{j+1})}$. By transitivity we have $D^{k_2}_{f(r_1)} < D^{k_j}_{s(r_j)}$ which is not possible by constraint (22). □

Lemma 21. *Let x, D, z satisfying (1)–(22), and let $i \in C$ be any customer. Let G^i be defined as before. Then the internal and external degree of every node $v \in G^i \setminus \{i^+, i^-\}$ are both equal to 1.*

Proof. Note firstly that every node in $G^i \setminus \{i^+, i^-\}$ has the same internal and external degrees, so G^i has an Eulerian path, starting at i^+ and finishing at i^-. If some node $v \in G^i$ has internal degree (and so external) greater or equal to 2, then the Eulerian path necessarily has a cycle containing node v, which is impossible by the previous lemma. □

Proposition 22. *Let x, D, z satisfying (1)–(22), and let $i \in C$ be any customer. Let G^i be defined as before. Then G^i is a path $(v_j)_{j=1}^J$ such that $v_1 = i^+, v_J = i^-$. Additionally, exists a tuple $(\mu_j)_{j=1}^J$ such that for each j, μ_j is the travel time of customer i from his origin to node v_j, and $\mu_j < \mu_{j+1}$ for each j.*

Proof. As proved in the previous lemmas, the graph G^i does not contain cycles and its nodes are of internal and external degree equal to 1, excepting i^+ and i^-. From here, we can deduce from Euler's Theorem for directed graphs that G^i is a simple path starting at i^+ and finishing at i^-. Thus, we can define

$$\mu_j = \begin{cases} D_{v_j} - D_{i^+} & \text{if } v_j \in N, \\ D^k_{s(r)} - D_{i^+} & \text{if } v_j = s(r) \in s(T) \text{ and } z^{ki}_{s(r)} = 1, \\ D^k_{f(r)} - D_{i^+} & \text{if } v_j = f(r) \in f(T) \text{ and } z^{ki}_{f(r)} = 1, \end{cases} \tag{C.3}$$

which represents exactly the travel time of customer i at each node in G^i. It's easy to see from constraints (9)–(15) and (22) that $\mu_j < \mu_{j+1}$. □

Proposition 23. *Let $x \in \{0,1\}, D \geqslant 0, z \in [0,1]$ satisfy constraints (1)–(23). Let us define, for each $r \in T, k \in M, i \in C$ a binary variable y^{ki}_r such that*

$$y^{ki}_r = z^{ki}_{f(r)} \quad \forall r \in T,\ \forall k \in M,\ \forall i \in C. \tag{C.4}$$

Then for all $k \in M, j \in V, i \in C, z^{ki}_j \in \{0,1\}$.

Proof. Let $k \in M$ be any vehicle, and let $i \in C$ be any customer. Note firstly that by constraint (21), for every node $v \in V$ not in the route of vehicle k, z^{ki}_v is equal to 0. Let now v be in the route of vehicle k, say $v = v_j$, being $P_k = (k^+, \ldots, v_{j-1}, v_j = v, v_{j+1}, \ldots, k^-)$ the route of vehicle k, and suppose that $z^{ki}_{v_{j-1}}$ take a binary value. Several cases may occur

Case 1: $v_{j-1} \in (N \setminus \{i^+, i-\}) \cup \{k^+\} \cup f(T)$: In this case, by constraint (17), we have that $z^{ki}_v = z^{ki}_{v_{j-1}}$, so z^{ki}_v takes a binary value, as well as $z^{ki}_{v_{j-1}}$.

Case 2: $v_{j-1} = i^+$: In such case, by constraint (18), $z^{ki}_{v_j} = 1$.

Case 3: $v_{j-1} = i^-$: In such case, by constraint (19), $z^{ki}_{v_j} = 0$.

Case 4: $v_{j-1} = s(r) \in s(T)$: In such case, $v_j = f(r)$, so by constraint (C.4), $z^{ki}_{v_j} = z^{ki}_{f(r)} = y^{ki}_r$ which takes values in $\{0,1\}$. □

References

Aldaihani, M., Dessouky, M., 2003. Hybrid scheduling methods for paratransit operations. Computers and Industrial Engineering 45, 75–96.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4, 238–252.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: A classification scheme and survey. TOP 15, 1–31.

Bianchessi, N., Righini, G., 2007. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations 34, 578–594.

Black, A., 1995. Urban Mass Transportation Planning. McGraw-Hill Series in Transportation, New York, USA.

Bodin, L., Golden, B., Assad, A., Ball, M., 1983. Routing and scheduling of vehicles and crews the state of the art. Computers and Operations Research 10, 69–211.

Codato, G., Fischetti, M., 2004. Combinatorial benders cuts. In: Nemhauser, G.L., Bienstock, D. (Eds.), IPCO of Lecture Notes in Computer Science, vol. 3064. Springer, pp. 178–195.

Contardo, C., 2005. Formulación y solución de un problema de ruteo de vehículos con demanda variable en tiempo real, trasbordos y ventanas de tiempo. Memoria para optar al título de ingeniero civil matemático, Universidad de Chile, Santiago de Chile.

Contardo, C., 2008. Personal website. <http://www.crt.umontreal.ca/~ccontard>.

Cordeau, J.-F., 2006. A branch and cut algorithm for the dial-a-ride problem. Operations Research 54, 573–586.

Cordeau, J.-F., Laporte, G., 2003a. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B 37, 579–594.

Cordeau, J.-F., Laporte, G., 2003b. The dial-a-ride problem (darp): Variants, modeling issues and algorithms. 4OR-Quartely Journal of the Belgian, French, and Italian Operation Research Societies 1, 89–101.

Cortés, C.E., 2003. High coverage point-to-point transit (hcppt): A new design concept and simulation–evaluation of operational schemes. Ph.D. Thesis, Ph.D. in Civil Engineering, Université of California, Irvine.

Cortés, C.E., Jayakrishnan, R., 2002. Design and operational concepts of a high coverage point-to-point transit system. Transportation Research Record 1783, 178–187.

Crainic, T.G., Malucelli, F., Nonato, M., 2001. Flexible many-to-few + few-to-many = an almost personalized transit system. In: TRISTAN IV. Sao Miguel Azores Islands, pp. 435–440.

Cullen, F.H., Jarvis, J.J., Ratliff, H.D., 1981. Set partitioning based heuristics for interactive routing. Networks 11, 125–143.

Desrochers, M., Lenstra, J., Savelsbergh, M.W.P., Soumis, F., 1988. Vehicle routing with time windows: Optimization and approximation. In: Golden, B., Assad, A. (Eds.), Vehicle Routing Methods and Studies. North-Holland, Amsterdam, pp. 65–84.

Desrosiers, J., Dumas, Y., Solomon, M., Soumis, F., 1995. Time constrained routing and scheduling. In: Ball, M., Magnanti, T., Monma, C., Nemhauser, G. (Eds.), Handbooks in Operations Research of Network Routing, vol. 8. Elsevier Science Publishers B.V., pp. 35–139.

Desrosiers, J., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. American Journal of Mathematical and Management Sciences 6 (3–4), 301–325.

Dumas, Y., Desrosiers, J., Soumis, F., 1989. Large scale multi-vehicle dial-a-ride problems. Technical Report Cahiers du GERAD G-89-30, École des Hautes Études Commerciales, Montréal, Canada.

Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. European Journal of Operational Research 54, 7–22.

Dumitrescu, I., 2005. Polyhedral results for the pickup and delivery travelling salesman problem. Technical Report CRT-2005-27, Centre de Recherche Sur Les Transports, Université de Montréal.

Gendreau, M., Guertin, F., Potvin, J., Séguin, R., 1998. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Technical Report CRT-98-10, Centre de Recherche Sur Les Transports, Université de Montréal.

Haghani, A., Jung, S., 2005. A dynamic vehicle routing problem with time-dependent travel times. Computers & Operations Research 32, 2959–2986.

Hickman, M., Blume, K., 2000. A method for scheduling integrated transit service. In: Eighth International Conference on Computer-Aided Scheduling of public Transport (CASPT). Berlin, Germany.

Horn, M.E.T., 2002a. Fleet scheduling and dispatching for demand-responsive passenger services. Transportation Research C 10, 35–63.

Horn, M.E.T., 2002b. Multi-modal and demand-responsive passenger transport systems: A modelling framework with embedded control systems. Transportation Research A 36 (2), 167–188.

Ioachim, I., Desrosiers, J., Dumas, Y., Salomon, M.M., Villeneuve, D., 1995. A request clustering algorithm for door-to-door handicapped transportation. Transportation Science 29 (1), 63–78.

Jaw, J.-J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.M., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research B 20B (3), 243–257.

Jih, W., Yun-jen, J., 1999. Dynamic vehicle routing using hybrid genetic algorithms. In: Proceeding of the IEEE International Conference on Robotics & Automation. Detroit, Michigan, USA, pp. 453–458.

Lau, H.C., Liang, Z., 2002. Pickup and delivery with time windows: Algorithms and test case generation. International Journal on Artificial Intelligence Tools 11, 455–472.

Le Bouthillier, A., Crainic, T.G., 2005. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. Computers & Operations Research 32, 1685–1708.

Li, F., Golden, B., Wasil, E., 2005. Very large-scale vehicle routing: new test problems, algorithms, and results. Computers & Operations Research 32, 1165–1179.

Liaw, C., White, C., Bander, J., 1996. A decision support system for the bimodal dial-a-ride problem. IEEE Transactions on System, Man and Cybernetics 26, 552–565.

Lu, Q., Dessouky, M., 2004. An exact algorithm for the multiple vehicle pickup and delivery problem. Transportation Science 38 (4), 503–514.

Madsen, O.B.G., Raven, H.F., Rygaard, J.M., 1995. A heuristics algorithm for a dial- a-ride problem with time windows, multiple capacities, and multiple objectives. Annals of Operations Research 60, 193–208.
Malucelli, F., Nonato, M., Pallottino, S., 1999. Demand adaptive systems: Some proposals on flexible transit. In: Crainic et al. (Eds.), Operational Research in Industry. McMillan Press, London, UK, pp. 157–182.
Mitrović-Minić, S., Laporte, G., 2006. The pickup and delivery problem with time windows and transshipment. INFOR 44, 217–227.
Nanry, W.P., Barnes, J.W., 2000. Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research B 34, 107–121.
Osman, M., Abo-Sinna, M., Mousa, A., 2005. An effective genetic algorithm approach to multiobjective routing problems (morps). Applied Mathematics and Computation 163, 769–781.
Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many inmediate request dial-a-ride problem. Transportation Science 14 (2).
Psaraftis, H.N., 1983. An exact algorithm for the single-vehicle may-to-many dial-a-ride problem with time windows. Transportation Science 17 (3).
Psaraftis, H.N., 1988. Dynamic vehicle routing problems. In: Golden, B., Assad, A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam, pp. 223–248.
Quadrifoglio, L., Hall, R., Dessouky, M., 2006. Performance and design of mobility allowance shuttle transit (mast) services: Bounds on the maximum longitudinal velocity. Transportation Science 40, 351–363.
Roy, S., Rousseau, J., Lapalme, G., Ferland, J., 1984. Routing and scheduling of transportation services for the disabled: Summary report. Technical Report CRT-473A, Centre de Recherche Sur Les Transports, Université de Montréal, Montréal, Canada.
Ruland, K.S., Rodin, E.Y., 1997. The pickup and delivery problem: Faces and branch-and-cut algorithm. Computers & Mathematics with Applications 33 (12), 1–13.
Savelsbergh, M.W.P., Sol, M., 1995. The general pickup and delivery problem. Transportation Science 29 (1).
Savelsbergh, M.W.P., Sol, M., 1998. Drive: dynamic routing of independent vehicles. Operations Research 46 (4), 474–490.
Sexton, T.R., Bodin, L.D., 1985a. Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. Transportation Science 19, 378–410.
Sexton, T.R., Bodin, L.D., 1985b. Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing. Transportation Science 19, 411–435.
Solomon, M., Desrosiers, J., 1988. Time window constrained routing and scheduling problems. Transportation Science 22 (1), 1–13.
Tarantilis, C., Ioannou, G., Prastacos, G., 2005. Advanced vehicle routing algorithms for complex operations management problems. Journal of Food Engineering 70, 455–471.
Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped person transportation problem. Transportation Science 31, 60–71.
Toth, P., Vigo, D., 2002. The Vehicle Routing Problem. Society of Industrial and Applied Mathematics (SIAM).
Xu, H., Chen, Z.-L., Rajagopal, S., Arunapuram, S., 2003. Solving a practical pickup and delivery problem. Transportation Science 37, 347–364.